Lab 8: The Curses Library

The *curses* library defines a number of functions that give you precise control over displaying characters on a computer screen. The name *curses* was apparently derived from the term “cursor” by someone with a crude sense of humor. There is also a newer version of the library called *ncurses*, short for “new curses”, that is available.

To use the *curses* library you need to include the *curses* header file:

```
#include <curses.h>
```

and explicitly link it during the compiling process. Standard libraries are linked automatically, but because the *curses* library is not always used it has to be explicitly linked. To compile a program that includes the *curses* library type:

```
g++ prog.cpp -lcurses
```

where the `-lcurses` directs the `g++` compiler to use the *curses* library.

There are a few basic commands that are a part of any program that uses the *curses* library:

- **WINDOW * variableName**;
  Declares a variable of type pointer-to *WINDOW*. A *WINDOW* is a C++ *struct*, and a pointer is essentially a reference. You don’t need to know exactly how these work in order to complete your lab.

- **variableName = initscr();**
  Connects the *WINDOW* variable to the screen

- **clear();**
  This function clears the screen.

- **refresh();**
  This function redraws the screen. It must be used every time you want something new to be displayed.

- **endwin();**
  This function ends *curses* control of the screen. It should go at the end of the program. If you forget to include it at the end of the program, when the program exits the screen will no longer respond properly and you will have to exit and log back on to the system.

Note how each of these commands is used in the sample program fragment given below.

There are three simple functions that help you place and remove characters on the screen:

- **move( r, c );**
  This function moves the cursor to row `r` and column `c`. `r` and `c` can be integers, integer variables or integer expressions.
• `insch(ch)`;
  This function places the character `ch` at the cursor’s current position. `ch` can be a character literal such as ‘R’ or a character variable.

• `delch()`;
  This function deletes the character at the cursor’s current location.

For example, the code:

```c
int row = 5, column = 10;
move(row,column);
delch();
insch('X');
refresh();
```

removes whatever character is currently at the location (row 5, column 10) and replaces it with the character X. Note that without the `refresh()` command the changes wouldn’t show up on the screen. Also note that the preliminary commands (defining a variable of type `WINDOW`, initializing the screen, etc.) must be done first.